PC Support Advisor

The Essential Resource for PC Support Professionals

Update 65

On This Month's Utility Disk

Windows icon-based program launcher.
PC utility to unpack Mac Stuffit archives.
Sample WordPerfect 6.0 for DOS coaches, including source code.
PC disk IO performance monitor.

"WordPerfect can survive" - says Robert Schifreen.

I used to share an office with a magazine called Educational Computing. This was in the days when computers had only recently been introduced into schools and "computer studies" meant "computer programming". In those days, you spent your time in the computing class by learning Basic, Fortran, a bit of Algol, and how to count in binary.

Nowadays, it's changed a lot. Children start using computers at school when they're very young and, in all probability, they also use a computer at home or at their parents' office, too. Where once they learned how to program the machine, now they learn how to use the thing. Basic and Fortran have given way to Excel and PageMaker.

I'll never forget the time when I saw a friend of mine sit down at her mother's computer, fire up WordPerfect 5.1 and type her name, even stopping to correct errors with the Delete key. That was when she was two and a half years old. Now, at four and a half, she'd rather draw pictures with Windows Paintbrush than a box of wax crayons.

I ran into the new CEO of WordPerfect Corporation recently at a press conference and I couldn't help telling him the story. He seemed remarkably unsurprised. WordPerfect has long since realised, he told me, that children make ideal beta testers for software. Children use software by instinct, rather than by logic, so the best way to find out whether you've designed your latest menu structure properly is to drag an eight year old off the street, sit him or her down at a PC, and give them a list of tasks to perform. If they have difficulty doing the tasks, you juggle some of the menu options and try again with some new children.

This all makes a great deal of sense. So why, then, has WordPerfect been having such a hard time recently? Why aren't users rushing to WordPerfect 6.0 for DOS? And why is the Windows version so slow and buggy? Why do I never see any "secretary with WordPerfect for Windows skills wanted" adverts in the newspapers? And why do all the journalists who used to submit their work to me in WP 5.1 for DOS format now send me Microsoft Word for Windows files?

If WordPerfect is to survive, it needs to listen to its large band of loyal users. With its new top-level management team in place, let's hope that the company admits its recent mistakes and does something to remedy them. WordPerfect for Windows 5.1 and 5.2 just didn't work, yet the company managed to produce figures that suggested it was outselling Word for Windows and was a better product.

If the developers can't get it right, hype and marketing won't work. Someone needs to realise this before it's too late.

PCSA

PC Support *Advisor*

PC Support *Advisor* is a publication of **International Technology Publishing**.

ISSN: 1031-3966

Editor: Robert Schifreen

Editor in Chief: Les Bell

Publisher: Chris Carvan

Email:
hex@cix.compulink.co.uk
CompuServe: 100016,301

International Technology Publishing Inc.
North American Subscriptions
Suite 200
555 De Haro Street
San Fransisco, CA94107
USA

Telephone: (415) 255 1295
Facsimile: (415) 255 8496

International Technology Publishing Inc.
European Subscriptions
13 Callcott Street
London
W8 7SU
United Kingdom

Telephone: (071) 724 9306
Facsimile: (071) 706 3296

International Technology Publishing Pty Ltd.
SE Asian Subscriptions
138 Cecil Street
Cecil Court #13-03
Singapore
0106

Telephone: (65) 227 4187
Facsimile: (65) 227 2885

International Technology Publishing Pty Ltd.
Australasian Subscriptions
Suite 511
185 Elizabeth Street
Sydney, NSW, 2000
Australia

Telephone: (02) 261 4683
Facsimile:(02) 261 4741

Understanding PCMCIA

Dave Stott presents the first of a two-part description of PCMCIA technology, which is a major component of modern laptop machines.

The growth in the popularity of mobile computers such as Notebooks, Sub-Notebooks, Palmtops and PDAs (Personal Digital Assistants) has fuelled the need for a new type of add-on card and associated interface.

Desktop PCs are available with a variety of different bus architectures and chassis systems that can accommodate full sized add-on cards up to a maximum of around 12 inches in length. However the portable computer, due to its small size, is severely restricted in respect of its expansion capabilities. This means that standard bus architectures such as ISA, EISA or MCA are totally inappropriate for such machines.

In the past, many manufacturers produced their own proprietary designs to overcome the problem of providing expansion cards for such systems. Sadly, this resulted in a wide variety of formats which naturally led to incompatibility.

Fortunately an international standard has emerged which addresses the problem of add-ons for mobile computers. The Personal Computer Memory Card Interface Association is the body responsible for specifying and developing the standard simply known as PCMCIA.

In this two-part feature I will cover all the major aspects of this technology which is set to become a significant factor for the mobile computer user. Part one covers the PCMCIA standard itself and associated memory cards. Part two will cover the wide range of other PCMCIA cards currently available.

Background

The PCMCIA was formed in June 1989 with the objective of promoting and educating the market in the then new technology of memory cards for use in portable computers. Since 1990 PCMCIA has worked in conjunction with JEIDA (the Japan Electronics Industry Development Association) to deliver an internationally recognised standard. Such a standard is extremely important to end-users of PCMCIA technology as it assures full compatibility between products from various vendors of all systems offering PCMCIA-compliant interfaces.

Currently the association has over 300 members from a broad range of backgrounds including memory card manufacturers, computer manufacturers and software specialists. All the major computer manufacturers including IBM, Hewlett-Packard, Apple, Toshiba and Sony are active members of the association, as are the major software developers such as Microsoft and Lotus.

Initially this technology was confined to the development of memory cards but, since the foundation of the PCMCIA, the scope has widened to cover a variety of different add-ons and peripherals to meet the needs of the portable PC user. These non-memory specific cards are usually referred to as "I/O cards". They are designed to provide the mobile user with the same sort of facilities and functionality as the expansion cards available for desktop PCs. So, common PC expansion peripherals such as modems, network adapters, sound cards and, more recently, even hard disks, have been developed to make use of the PCMCIA interface. This has put the capabilities of the portable PC almost on a par with its desk bound equivalent.

The PC Card

Cards designed to meet the PCMCIA specifications (known, rather confusingly, as PC Cards) are the size and shape of a typical credit card and offer a small and convenient approach to expanding the capabilities of a mobile PC. They are based on solid state technology, very lightweight, highly reliable, very rugged and consume very little power.

Part of the PC Card specification is the definition of a Card Information Structure (CIS). This is a data structure consisting of small elements called "tupples" which can be examined by the host software to determine the capabilities of a particular card. Every card that complies to the PCMCIA standard contains information about the formatting and organisation of the data on the card.

Versions 1 And 2

The original PCMCIA specification known as Version 1 was introduced in August 1991 and was solely concerned with memory expansion cards and supported all memory types except DRAM. Whilst this proved to be an acceptable method of adding various types of memory to a portable PC its limitations soon led to the development of Version 2. The specification for PCMCIA Version 2 was released in September 1991.

The enhanced Version 2 specification included support for I/O devices, dual voltage cards, XIP (eXecute In Place) and increased support for flash-memory cards. This current standard has greatly increased the versatility of the interface and enabled manufacturers to produce a wide variety of peripherals in the form of PC Cards.

XIP means that code stored on a card executes on the card by mapping the card's memory into the computer's memory map. This speeds up program loading, as there's no need to copy the code from the card into the

computer's RAM. Also, there's no need for the computer to have massive amounts of its own RAM - whatever card is plugged into the machine becomes the computer's memory.

Version 2 of the standard is fully backwards compatible with Version 1 so that older PCMCIA cards can still be used.

Physical Features

The PCMCIA standard defines an interface slot consisting of a 68-pin connector in the host computer. These pins are arranged in two rows of 34. PC Cards have a female socket which directly mates with the male interface socket.

One unusual feature of this interface is that the standard specifies three different connection pin lengths, ie, the various pins on the connector are a variety of lengths. This ensures that when a card is inserted into the slot, the power is applied first, and when the card is removed the power is cut off last. The design of this feature is necessary to guarantee the safe and reliable operation of PC Cards which are being inserted and removed from what is in effect an electrically live socket.

Figure 1 lists all the signals and functions of the 68-pin PCMCIA interface. You will notice that some of the pins have a dual or alternative function. In all, 10 of the 68 pins have alternative functions. This is due to the differences between Version 1 and 2 of the standard which enable the interface to use I/O cards as opposed to just memory cards.

For example, when accessing a memory card, pin 16 is used as a RDY/BSY (Ready/Busy) signal, but this function changes to IREQ (Interrupt REQuest) when an I/O card is being used. Similarly, pin 18 becomes the peripheral voltage supply line; pin 33 detects when the I/O port is 16-bit; pins 44 and 45 become I/O read and write, respectively; pin 52 becomes the second peripheral voltage supply line; pin 60 becomes the input port acknowledge line; pin 61 becomes register select and I/O enable; pin 62 changes to audio-digital waveform; and pin 63 is used to identify changes to the card status.

A 26-bit address bus enables the interface to directly address up to a maximum of 64 MB. The data bus which is used for I/O operations may be either 8- or 16-bits wide, with the card activating pin 33 to signal 16-bit use to the system. It is also possible to work with 16-bit operations on 8-bit I/O ports by dividing the operation into two consecutive 8-bit operations, in the same way that PCs utilise the ISA bus structure.

Pin number	Standard function (alternative function)	Input/Output	Description
1	GND		Ground
2	D3	I/O	Data-bus bit 3
3	D4	I/O	Data-bus bit 4
4	D5	I/O	Data-bus bit 5
5	D6	I/O	Data-bus bit 6
6	D7	I/O	Data-bus bit 7
7	-CE1	I	Card Enable
8	A10	I	Address-bus bit 10
9	-OE	I	Output Enable
10	A11	I	Address-bus bit 11
11	A9	I	Address-bus bit 9
12	A8	I	Address-bus bit 8
13	A13	I	Address-bus bit 13
14	A14	I	Address-bus bit 14
15	-WE/PGM	I	Write Enable
16	+RDY/-BSY (-IREQ)	O	Ready/Busy or Interrupt Request
17	Vcc	Power	
18	Vpp1	Programming supply voltage 1	
19	A16	I	Address-bus bit 16
20	A15	I	Address-bus bit 15
21	A12	I	Address-bus bit 12
22	A7	I	Address-bus bit 7
23	A6	I	Address-bus bit 6
24	A5	I	Address-bus bit 5
25	A4	I	Address-bus bit 4
26	A3	I	Address-bus bit 3
27	A2	I	Address-bus bit 2
28	A1	I	Address-bus bit 1
29	A0	I	Address-bus bit 0
30	D0	I/O	Data-bus bit 0
31	D1	I/O	Data-bus bit 1
32	D2	I/O	Data-bus bit 2
33	+WP (-IOIS16)	O	Write Protect or I/O port is 16-bit
34	GND		Ground
35	GND		Ground
36	-CD1	O	Card Detect
37	D11	I/O	Data-bus bit 11
38	D12	I/O	Data-bus bit 12
39	D13	I/O	Data-bus bit 13
40	D14	I/O	Data-bus bit 14

Figure 1 - Signals and functions of the 68-pin PCMCIA interface.

Physical Guides

Physically the PCMCIA slot uses two guides to ensure correct alignment of the card's socket with the connector pins. These are 3.3mm wide (there are 25.4mm, or 2.54cm, to an inch). A small cut-out on the actual card acts as a key to maintain correct orientation of the card during insertion, and thus the card cannot be inserted the wrong way round. This avoids damage to either the interface slot or the card itself.

PCMCIA

The PCMCIA standard also defines the location and size of the write-protect switch as well as the internal backup battery for memory cards. Other card specific items such as vendor labelling and the use of the PCMCIA and PC Card logos are covered by the specification.

The cards themselves must conform to certain operating constraints. They must operate at between 0 and 55 degrees centigrade, survive storage temperatures of -20 to +65 degrees centigrade, and 0 to 95% humidity, non-condensing. In addition they must be capable of withstanding a shock equivalent to three successive drops from a height of 75cm (30 inches).

Types I, II And III

There are three PC Card types specified in the current PCMCIA standard. Each of these differs in terms of physical thickness and yet they all used the same 68-pin edge connector. All three card types measure 85.6mm by 54.0mm (see Figure 2). Type I cards are the thinnest and are a uniform 3.3mm thick. This thickness exactly matches the guides specified for PCMCIA slots. Both Type II and Type III cards conform to this dimension in order to fit in the slot guides. However, these cards are thicker in the centre, unlike Type I cards which are a uniform thickness. The Type I format is typically used for memory enhancement PC Cards as the components used can be relatively easily accommodated in their slim cases.

Type II cards are restricted to a 5 millimetre form factor at the centre and this allows greater scope for internal components. Typically Type II cards are used for I/O cards such as modems or LAN adapters. There are some Type I PC Card modems and LAN adapters available which, for example, use short drop cables to provide the necessary RJ11 telephone socket and RJ45 or BNC network sockets.

The thickest cards are the Type III which are 10.5mm at the centre. This extra depth is required for I/O functions that need more space, rotating media (hard disks) or wireless communications devices (cellular radio).

In order to accommodate the extra thickness of Type III cards, portable PC manufacturers often include two PCMCIA slots in their products. These two slots are usually stacked one above the other. This double socket arrangement allows either two Type I cards to be plugged in and used simultaneously or one Type II or III card to be plugged in by itself.

The Power Issue

Users of portable PCs are extremely conscious of the battery life which systems offer. This has accelerated the development of low voltage processors such as the AMD 386SXLV, the Cyrix SLC/e and the 486SLC from Intel. Consequently it is very important that, when in use, PC Cards should not place too many power demands on the host system.

With this aspect in mind the PCMCIA has been actively pursuing the adoption of low voltage technology in PC Cards. The original Version 1 specification only catered for cards operating at 5 volts but Version 2 included support for low power consumption cards operating at 3.3 volts.

When a low power consumption PC Card is inserted into a Version 2

41	D15	I/O	Data-bus bit 15
42	-CE2	I	Card Enable
43	RFSH	I	Refresh
44	RFU (-IORD)	I	Reserved for Future Use or I/O Read
45	RFU (-IOWR)	O	Reserved for Future Use or I/O Write
46	A17	I	Address-bus bit 17
47	A18	I	Address-bus bit 18
48	A19	I	Address-bus bit 19
49	A20	I	Address-bus bit 20
50	A21	I	Address-bus bit 21
51	Vcc	Power	
52	Vpp2	Programming supply voltage 2	
53	A22	I	Address-bus bit 22
54	A23	I	Address-bus bit 23
55	A24	I	Address-bus bit 24
56	A25	I	Address-bus bit 25
57	RFU		Reserved for Future Use
58	+RESET	I	Reset
59	-WAIT	O	Extend bus cycle
60	RFU (-INPACK)	O	Reserved for Future Use or Input Port Acknowledge
61	-REG	I	Register select
62	BVD2 (-SPKR)	O	Battery Voltage Detect 2 or audio-digital waveform
63	BVD1 (-STSCHG)	O	Battery Voltage Detect 1 or card status changed
64	D8	I/O	Data-bus bit 8
65	D9	I/O	Data-bus bit 9
66	D10	I/O	Data-bus bit 10
67	-CD2	O	Card Detect
68	GND		Ground

Standard Functions apply to memory cards and Alternative Functions apply to I/O cards such as modems, LAN adapters etc.

Figure 1 - Signals and functions of the 68-pin PCMCIA interface (Cont).

slot the operating voltage is initially set to 5 volts. This is done to ensure compatibility with older cards which can only operate at this voltage. The host system then reconfigures the PC Card to operate at 3.3 volts if it is capable of supporting it.

Socket Services

The PCMCIA standard not only covers the hardware specifications but also encompasses software. The software is hierarchical and there are two key elements defined called Socket Services and Card Services.

Socket Services is a BIOS level software interface that provides a method for accessing the PCMCIA slots of a computer. In a normal PC environment Socket Services are accessed via interrupt 1Ah.

There are two types of resources which Socket Services is designed to manage, namely adapters and sockets. The connection between the host system's internal architecture and the PCMCIA interface is known as an adapter. A socket is the physical 68-pin interface connector into which PC Cards are inserted and is accessed by the host via the adapter. Normally, a PC system will have just a single adapter but it is possible to have more than one. A single adapter can control multiple sockets on a host by using multiplexing techniques.

Within Socket Services there is a set of low-level system functions which application programs can call. These are grouped into the following categories: Non-specific; Adapter; Window; Socket; Card; Error Detection Code (EDC).

Non-specific functions are used to determine the number of adapters in the system and for establishing call-back routines. A call-back routine is used to establish whether a card's status has changed or when write errors are detected.

Adapter functions are used to determine the version number of the current Socket Services handler, to determine the current configuration of a particular adapter and to set the configuration of a specific adapter.

The Window functions are used for the manipulation and management of memory-space windows. These are an integral part of the Socket Services design. A window is used to map an area of a PC Card's memory or I/O space to the memory or I/O space in a host system. Windows can consist of zero or more pages, but if a window is paged then all the pages must be contiguous and of equal size. I/O spaced windows are not split up into pages.

Socket functions are very similar to Adapter functions and are used to establish the configuration of sockets (as in the 68-pin connector) as opposed to adapters.

The Card functions apply to the actual PC Cards themselves and are mainly used for controlling the read/write operations. Both single byte/word and multiple bytes/words read/write operations can be handled by these functions depending on the actual cards' capabilities.

Finally, certain memory cards have their own EDC generators which are used to self-check the integrity of the card's memory. The EDC functions determine the EDC capabilities of a particular memory card and will configure the card accordingly. Functions are also available to stop and start the EDC generated by a card.

Card Services

Card Services is a software management interface that allows the allocation of system resources (such as memory or interrupts) automatically once the Socket Services has detected that a PC Card has been inserted. It operates at a higher level than Socket Services and controls the operation of individual cards by sending transactions over the bus via Socket Services.

The main purpose of Card Services is to liaise between multiple clients (tasks running in the host system) and PC Cards, sockets and other system resources. All access to PC Card functions from clients are initially handled by Card Services, which then makes calls to Socket Services to effect the necessary operations. As mentioned previously a system can have multiple adapters along with multiple associated Card Services controlling them, however, only a single implementation of Card Services can be used on a system.

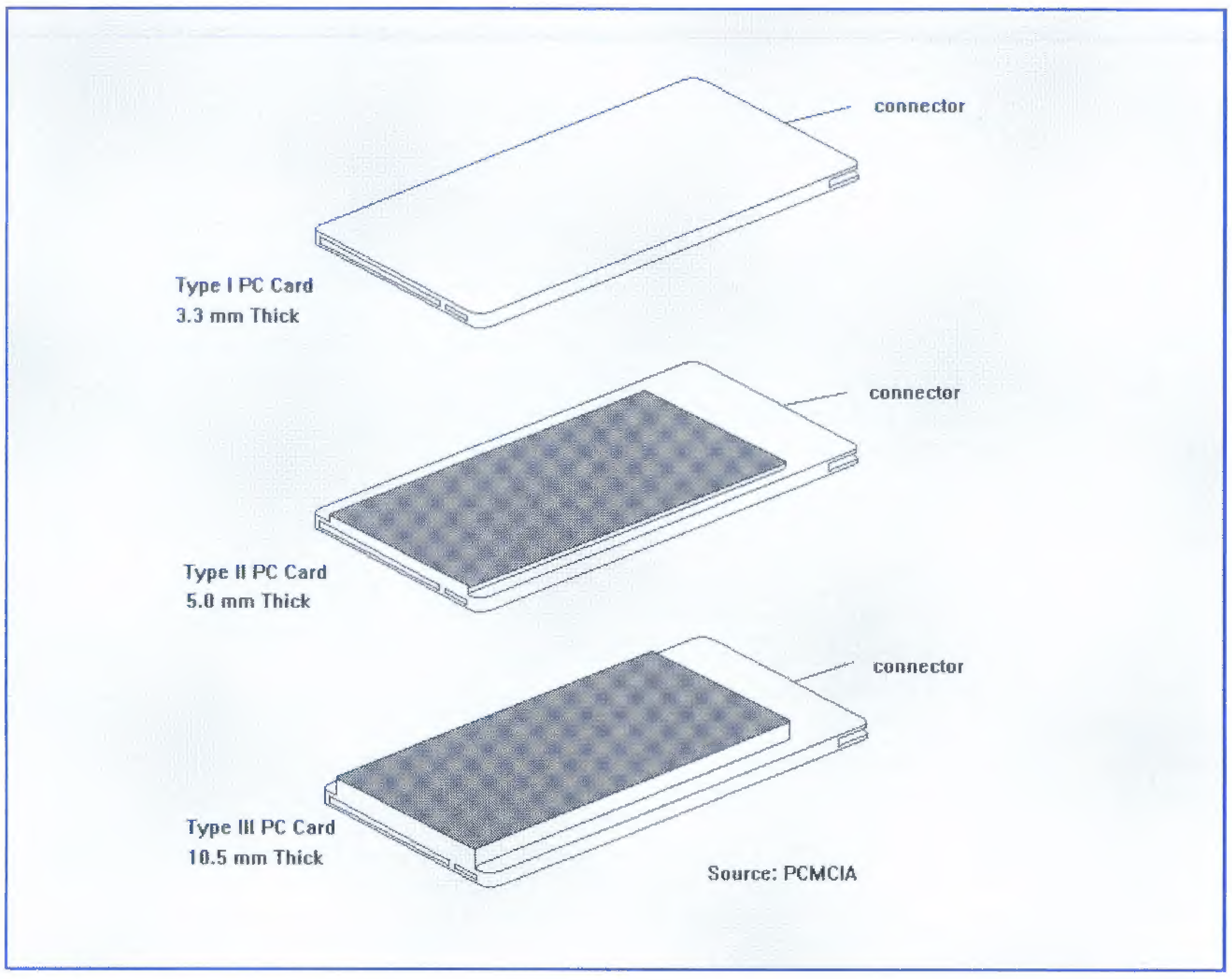

Figure 2 - There are three PC Card types specified in the current PCMCIA standard.

PCMCIA

Intel, an executive member of the PCMCIA, has defined its own software interface environment, the Exchangeable Card Architecture (ExCA) that replaces Card Services. ExCA is a superset of Card Services and the original model on which Card Services was based.

Plug And Play - QuickSwap

One of the key concepts of PCMCIA is that PC Cards can be readily swapped when different specific functions are required. Indeed the design of the physical elements of the specification are geared towards catering for multiple swapping of PC Cards and, in the case of memory cards, without data loss. Physically, PCMCIA equipped systems and PC Cards are expected to withstand up to 10,000 such card swaps during their lifetime.

This aspect highlights an area of interoperability which needs to be addressed, for although PCMCIA caters for "Hot Swapping" of PC Cards, how can the host system dynamically deal with different configurations associated with each type of card? Take, for example, a user accessing a PC Card modem from within a communications program. If the modem card is subsequently removed or replaced by, for instance, a LAN adapter card then the application originally accessing the modem card will fail because the resources that it expects to be available via the PCMCIA slot have disappeared.

As mentioned earlier, each PC Card has its own CIS (Card Information Structure) which determines the card's capabilities, but the host system has to be reconfigured in order to make use of these new functions when a card is swapped. Sometimes this reconfiguration will even involve re-booting the host system before the new card is correctly recognised and available for use.

The introduction of "Plug and Play" on desktop systems is designed to enable systems to automatically configure themselves depending on the resources made available by the expansion cards fitted. There already exists an API (Applications Programming Interface) for MS-DOS and Windows 3.1 called the Plug and Play Configuration Manager API which allows device drivers and other software to get information about how a particular piece of configurable hardware has been set up. The API was jointly defined by Microsoft and Intel and is documented in the "Plug and Play Kit for DOS and Windows", available from Intel or from the PLUGPLAY forum on CompuServe.

The Plug and Play Configuration Manager currently only supports PCI (Peripheral Component Interconnect), EISA and Plug and Play ISA bus architectures, as used by desktop systems. These are in effect "static" environments, where expansion cards are not hot-swapped but are installed with the power off. Resource allocation and configuration only takes place when the system boots up. This is completely different from the hot swapping capabilities expected of PCMCIA.

Using the original ExCA specification as a starting point, Intel has proposed an updated ExCA Release 1.5 specification, renamed "QuickSwap Specification Release 2.0, October 1993 (DRAFT)", to the PCMCIA for approval. In addition, the Plug and Play Configuration Manager has been enhanced to include support for PCMCIA for MS-DOS and Windows 3.1 and is expected to be forward-compatible with Windows 4.0 (Chicago). These developments mean that systems will be able to reconfigure themselves to take account of the new resources available when a card is swapped. To the end user this will all take place automatically and dynamically.

QuickSwap

Currently the PCMCIA is reviewing and modifying the QuickSwap 2.0 proposal and it is hoped that this specification, in some form, will be approved in the near future. End-users can probably expect to see new Plug and Play ready machines that support PCMCIA in Q2 94, depending on how quickly OEMs can integrate the new version of the configuration manager.

In the meantime a SIG (Special Interest Group), called the QuickSwap SIG, has been formed as an industry-wide group to promote the visibility and viability of the PCMCIA standards for Intel architecture based systems. The QuickSwap SIG's purpose is complementary to that of the PCMCIA.

Memory Cards

The provision of easily installed memory expansion is one of the primary uses of PCMCIA and the current specification provides support for a number of different memory card types including SRAM, EPROM, EEPROM, Flash Memory, OTPROM, and Mask ROM. Each of these memory card types has its own characteristics which determine their suitability for different purposes.

SRAM (Static RAM) cards are used for exchangeable memory applications. They have full read and write capabilities and are equipped with an internal battery. SRAM cards do not need to use the computer's processor in order to refresh the contents of their memory. SRAM cards will even retain their contents for up to half an hour whilst the internal battery is changed. Typically they are used as solid state disks and are available in a variety of capacities from 64 KB to 24 MB.

There are several different types of EPROM (Erasable Programmable ROM) cards available. Generally they differ in the way that the memory is erased. For example, UV-EPROM requires exposure to ultraviolet light whilst EEPROM (Electrically Erasable PROM) can be erased by use of a specific electric current. They are mainly used for applications requiring an area of memory that can be altered, but where read operations are far more frequent than write operations. Typical capacities available range from 256 KB to 1 MB.

Flash Memory is an emerging technology offering read/write capability without the need for an internal battery to constantly refresh memory contents. When in use the card derives electrical power from the PC via the PCMCIA slot. Unfortunately the architecture of Flash Memory cards

means that it is not usually possible to erase only part of the Flash Memory and effectively the memory has to be reformatted in order to change data already stored on it. Therefore. this technology is best used in situations where data is only modified occasionally.

Although initially it was considerably more expensive than SRAM, Flash Memory is gaining popularity and Intel already produces cards with capacities up to 4 MB.

The distribution of software is seen as an ideal application for PC Card memory and there are two formats providing a solution to this problem. OTPROM (One Time Programmable Read Only Memory) can be programmed in the same way as EPROM with specific information. However, its design means that it cannot be changed by either exposure to UV or by a change in electrical voltage.

As an alternative there is Mask ROM where the information to be stored is directly etched onto the card's memory chips. This process has to be done during the manufacturing process and hence it is really only economical when large production volumes are involved. Mask ROM cards offer very fast access to a large amount of memory. At present storage capacity for Mask ROM cards is up to a maximum of 32 MB.

One key element of PCMCIA Version 2 is XIP which enables software to be executed within the memory of the card itself, rather than copy the software to the hosts main memory for execution. This facility is especially useful when the host has limited internal memory as it conserves such resources for data storage.

The notable omission from the PCMCIA memory card standard is DRAM. There are two reasons for this. First because DRAM memory cards require 88-pin connections to the host. Second, because they are designed to be permanent memory enhancements in the same way that SIMMs or SIPPs are utilised in desktop PCs. Therefore, DRAM memory cards are not suitable candidates for PCMCIA which assumes that cards are likely to be swapped in and out, possibly on a frequent basis.

PCMCIA On The Desktop

To many people PCMCIA technology is perceived to be solely related to mobile computing. However, this is not necessarily the case. There are occasions when it is extremely useful to have PCMCIA slots available on desktop systems.

Software houses which are targeting the increasingly popular mobile market can utilise the power of their desktop platforms for software development. Desktop systems with PCMCIA capability can be used as a very effective "bridge" to mobile systems, making the exchange of information as simple as swapping PC Cards. The adoption of PCMCIA by manufacturers of desktop systems will enable them to build much smaller system units. Future desktop systems will have a sleek chassis and yet will be capable of considerable expansion due to the wealth of PC Cards available.

There are several ways that you can make PCMCIA technology available on your existing desktop system. By far the cheapest solution is to buy a PCMCIA expansion card. This can be a standard ISA bus card with a single Type I, II or III PCMCIA slot in the rear plate. This approach is fine if you are unlikely to be swapping cards frequently as to do so you will need easy access to the rear of the system unit.

A more convenient solution is a PCMCIA drive. These are available as both external and internal units. External drives either use a bus based expansion card or the standard parallel port as an interface. Obviously external drives which interface via the parallel port have the added advantage that they can be moved from one system to another relatively easily.

Internal PCMCIA drives are generally based on the standard 3.5 inch drive form factor. Once installed they are as easy to use as floppy disk drives with an indicator which shows the battery status of the card being used and a busy light to show when the card is being accessed. Some drives also have an eject button.

SCM Microsystems Inc of Los Gatos, California (Tel: +1 408 395 9292) supplies a comprehensive range of products under the name SwapBox to provide PCMCIA on the desktop. Typical prices are $139 for the MMCD-V ISA board solution up to $399 for the MMCD-SD which provides both a rear socket drive and a 3.5 inch internally mounted PC Card drive. Both products cater for all three PCMCIA card types.

Conclusion

The mobile computing market is growing at an enormous rate, much faster than the current desktop section of the market. Dataquest estimates that shipments of mobile systems will increase from the 5.1 million units in 1992 to over 16 million units in 1996.

PCMCIA is already establishing itself as the preferred technology for offering expansion capabilities in the mobile computing arena. The percentage of mobile systems sold with PCMCIA support is expected to reach 14% in 1994. The technology is also gaining ground in other areas, indeed Nokia has recently launched a portable digital phone with a PCMCIA Type I slot built in.

In part two of this feature [*in PCSA 66 - Ed*] I will take a look at the wealth of PCMCIA cards currently available to satisfy the needs of the mobile user.

PCSA

The Author

Dave Stott has worked in the computer industry for nearly 25 years. He is a freelance writer and IT consultant.

How To Install SCSI Devices

SCSI devices provide a fairly standard way of attaching high-speed, high-capacity drives and other devices to PCs. Julian Moss explains everything you need to know about linking PCs to SCSI.

Installing SCSI devices is a black art. There isn't one person in the world who could walk into your office one morning and guarantee to get a particular SCSI setup working by lunch time. So there's no need to feel embarrassed if the key to successful SCSI configuration seems constantly to elude you.

However, there's a lot you can do to improve your chances of success: like making sure you observe certain immutable principles that, if ignored, will almost guarantee failure. If you make as much effort as you can to do things right, your SCSI devices won't necessarily work first time, but at least you'll spend a lot less time in tiresome troubleshooting and trial and error.

SCSI Basics

SCSI devices are linked by means of the SCSI bus. A bus is a means of transporting data and control signals from one place to another. The SCSI bus fulfils a similar role to the expansion bus in a PC, though in operation it has more in common with a bus type network such as Ethernet.

A SCSI bus normally has up to 8 devices attached to it. The devices are linked using short lengths of cable from one device to another, forming a chain. The order in which devices are chained together is not important.

Every SCSI device has its own ID. This is a value in the range 0 to 7. The ID must be set by you, and its value is important as each device ID on a SCSI chain must be unique.

An ID may also be required to be a specific value, or one of a limited range of allowable values, in order for the device to function as intended. Check the device documentation and that of its driver software to see if this is the case. Ideally, check before you buy as there's no point in buying two devices for use with the same PC, each of which will only work if they have a device ID of 6.

SCSI devices are rarely shipped with the correct ID by default - if they are it's probably just a coincidence - so you can't assume that if you leave the jumpers or DIP switches alone a device ought to work.

There is a second component of the device identification, which is known as the Logical Unit Number or LUN. This can also have a value from 0 to 7. You will probably not need to worry about the LUN. The default setting of 0 should work in most cases, as long as the ID of each device is different.

Host Adaptor

Every SCSI bus must have attached to it one device called the host adapter. This is usually a plug-in board, though some high-specification PCs have it incorporated into the motherboard. The host adapter is sometimes called a SCSI controller.

It is theoretically possible to have more than one host adapter attached to a single SCSI bus, using different LUNs. One would be the master, and have a LUN of 0. However, SCSI can be difficult enough to set up and get working without this additional complexity, so to keep things reasonably straightforward we will assume that you can't.

To summarize, then, a SCSI bus consists of a chain of devices which includes one host adapter and up to seven SCSI peripherals. Each device must have different IDs. Specific devices may need to have specific IDs. The host adapter almost always has an ID of 7.

"Every SCSI bus must have attached to it one device called the host adapter. This is usually a plug-in board, though some high-specification PCs have it incorporated into the motherboard. The host adapter is sometimes called a SCSI controller."

Termination And Cabling

One aspect of setting up a SCSI installation is fundamentally important, and that's termination. This probably causes more problems than everything else combined. If the SCSI bus isn't correctly terminated then it will work poorly, or not at all.

Terminators have two important tasks to perform. First, they must pull up the signal lines on the SCSI bus to a logic high voltage level (between 2.5V and 5.25V.) Second, they provide impedance matching of the bus cabling.

The wires in a SCSI cable are transmission lines. The electrical signals that travel along them travel in waves. When waves hit the unterminated end of a transmission line, they are reflected back towards the source. The terminator absorbs the unwanted energy so that this doesn't happen.

The effect of reflected waves is to interfere with the signals you want. The result, as far as a device is concerned, depends on exactly where in the SCSI chain it is. At a point where the reflected waves are in phase with the wanted signals, the waves will be added together. Where they are out of phase they will tend to cancel out. Most of the time the device will be somewhere in between, and will receive a distorted signal. In no case, except perhaps the first, will it receive a signal it can read.

If you have a SCSI setup in which problems appear or disappear - or change - when you swap devices around on the chain or change the lengths of cables, you very likely have a termination problem. It's worth noting here that some host adapters are more sensitive to termination problems than others - Adaptec's AHA-1542C is an example of this - so correct termination is paramount.

Fortunately the rules are very simple. If the SCSI chain begins with device A, and ends with device Z, then devices A and Z must be terminated. If there are any other devices in between - in other words, with cables going both in and out - they must not be terminated. The rules apply equally to host adapters as to peripherals.

Given that you can have both external and internal SCSI peripherals, there are three possible combinations. The situation in which you have only internal devices is shown in Figure 1. In this configuration, the host adapter and the last device in the chain must be terminated. If the middle device(s) are shipped with terminators fitted, they must be removed.

"Though we talk glibly of adding and removing terminators, how you achieve this in practice varies from one device to another. There are three basic methods of termination. The most common - used by SCSI devices since the earliest days - is passive termination."

"Most SCSI devices use what are known as single-ended drivers, which means that the electrical signals used vary between a nominal 0V and 5V. A few devices have differential drivers, in which the signal levels are positive and negative."

The second situation is depicted in Figure 2. Here, all the SCSI peripherals are external. Once again, the host adapter and the last device in the chain must have terminators on.

The third example, shown in Figure 3, is the most complex, though the rules are still the same. Here, we have both internal and external devices. The internal devices are attached, as before, by ribbon cable, and the last one is terminated. The external devices are attached to the connector on the back of the PC, and the last one of these is also terminated. The host adapter in this instance is effectively in the middle, and so its terminator should be removed.

Cabling

Another point to watch is that the cabling is within specification. The total length of a standard SCSI bus is six metres (19.7 feet). However, the closer you get to the maximum length the more critical factors like correct termination become. Use the shortest possible cables at all times. However, external cables should have a length of at least 30cm (one foot.)

Ribbon cable is used for internal wiring. Bare devices have only a single SCSI connector, and the chain is formed by having multiple connectors on the cable.

External connections are generally made using screened cable and connectors. Devices have two SCSI sockets, which are used to attach either two cables (in and out), or one cable and a terminator.

SCSI Devices

"A few selfish SCSI devices have only one socket, and a built-in terminator permanently wired in place. These devices can only be used at the end of a chain: if you want to connect two of them then you can't. This type of device is fortunately becoming rare."

It is possible to use ribbon cable for external wiring as long as the lengths are short and it is kept away from sources of interference like power leads and power supplies. Using IDC (insulation displacement connectors) fittings and a vice it is easy to make up your own SCSI cables using ribbon cable.

Most SCSI devices use what are known as single-ended drivers, which means that the electrical signals used vary between a nominal 0V and 5V. A few devices have differential drivers, in which the signal levels are positive and negative.

Differential SCSI drivers are more resistant to interference. This means that the maximum length of the bus can be extended to 25 metres (82 feet.) Note that damage will result from trying to mix differential and single-ended devices on the same bus.

Types Of Termination

Though we talk glibly of adding and removing terminators, how you achieve this in practice varies from one device to another. There are three basic methods of termination. The most common - used by SCSI devices since the earliest days - is passive termination. This is electrically the most simple method, but has a fairly high power dissipation.

More modern devices usually offer the option of active termination. The advantage of this is that it provides a closer match to the impedance of the cable. The closer match minimizes reflections. Active termination also has a better immunity to noise. It is usually required for higher performance SCSI-II configurations. Consult your device documentation to see if active termination is possible.

Active and passive termination can both be used on the same SCSI bus, and will give better results than if only passive termination is used.

Differential SCSI devices use yet another type of termination, which is essentially passive.

Having determined the method of termination to use, the next step is to identify the terminators. For external devices they usually take the form of a plug like the one on the end of the cable, and are simply fitted to the unused one of the pair of sockets on the last device in the chain. Some devices, though, have a built-in terminator which can be turned on and off - often using a switch accessible from the back.

A few selfish SCSI devices have only one socket, and a built-in terminator permanently wired in place. These devices can only be used at the end of a chain: if you want to connect two of them then you can't. This type of device is fortunately becoming rare. Though it may seem a ridiculous piece of design, it is only quite recently that PCs using more than one SCSI peripheral have become commonplace so for a long time for many users it did not prove to be a problem.

One thing to watch for is the double-ended terminator. This has a male connector on one side and a female on the other. This is not intended to be fitted between the cable and the socket on the device. It is designed to

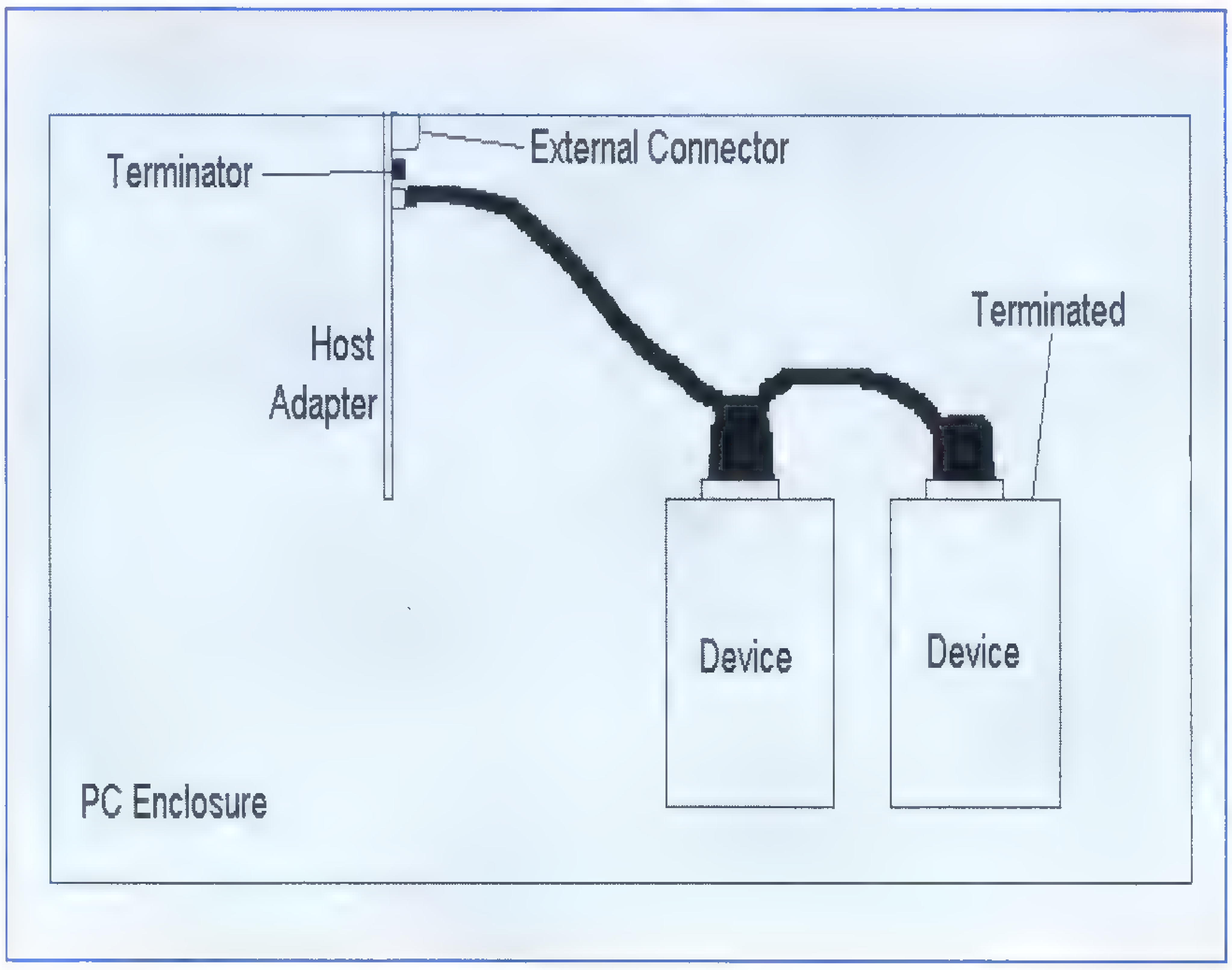

Figure 1 - Example SCSI configuration where all SCSI devices are internal to the PC.

be used as an ordinary external terminator. However, the reverse side means that if you should need for any reason to remove the last device in the chain, you can terminate the free end of the cable instead of having to disconnect it from the previous device. If the terminator is fitted in-line it will function as an attenuator, and the device it is attached to is very unlikely to work.

Bare devices intended for installation inside the PC have their terminators built into the device logic board. A common method is to use resistor packs installed in sockets on the board. They are shiny epoxy-coated components, often black in colour, and are narrow and about an inch long with a single row of pins. Usually, there are three of them. If the device is not at the end of the SCSI chain then these resistor packs should be removed.

Other devices may use switches or jumpers to electrically switch the terminator in and out of circuit. You'll need to refer to the device documentation to identify the correct settings.

"Host adapters offer a range of capabilities. The basic models have no built-in ROM and are suitable only for non-DOS devices like CD-ROM or tape drives which are accessible only through device drivers. Others have a boot ROM which implements the BIOS hard disk functions."

Terminator Power

One last point to check in relation to termination is which devices are supplying terminator power. You will recall that one of the functions of the terminators is to pull the signal lines up to a logic high voltage level. To do that, power must be supplied. This is terminator power.

The rules are that at least one, and no more than four SCSI devices should supply terminator power to the bus. This is normally enabled by means of a jumper or switch on the device. By default, the host adapter card will be set to provide terminator power; other devices generally are not.

If you are using one of the pocket parallel port SCSI adapters then you will probably need to set one of the drives to supply terminator power, as the pocket adapter does not have enough power to do so itself.

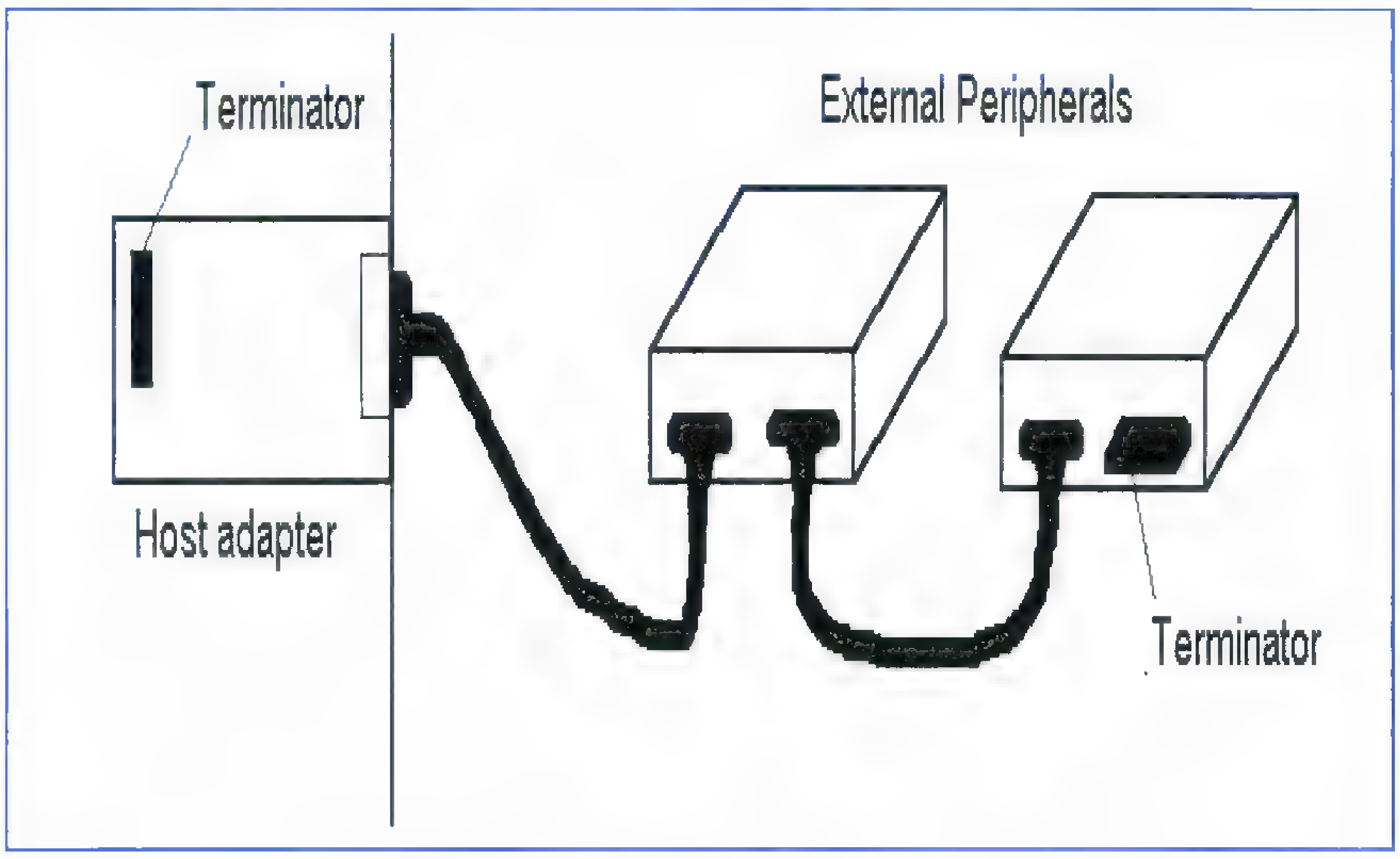

Figure 2 - Example SCSI configuration where all SCSI devices are external.

The Host Adapter

The host adapter is the channel through which the PC and its software communicates its requirements with the devices on the SCSI bus. Its capabilities are determined by those of its chip set and on-board firmware. It is essential that the host adapter supports all the SCSI features expected by a device. This is not always the case. Though incompatibilities are less of a problem than they once were, it is always a good idea to check with the vendor of either (or both) the device and the host adapter card, to make sure that they will work with each other.

To get round these compatibility problems, some vendors sell devices bundled with a host adapter card as a matched pair. A few may even go so far as to say that they will only support the device if the supplied adapter is used. This can lead to the situation in which a PC has two or more host adapters installed, each with perhaps a single device attached. However, it may well be possible to use the device with a different host adapter. A simple phone call to the manufacturer may be all that it takes to check.

As a general rule, older host adapter cards tend to have more compatibility problems, and cheaper ones are often more restrictive about what they can be used with. For example, the Seagate ST01 only supports specific hard drives.

SCSI Devices

"Check the host adapter's SCSI ID. This should be 7. If the host adapter will not be at one end of the SCSI bus, remove or disable the terminators. And unless you have a good reason for it not to be, make sure terminator power is enabled."

Adaptec

If you are starting from scratch with a new host adapter then there is a lot to be said for buying an Adaptec. Adaptec's host adapters have become something of an industry standard, in much the same way that Western Digital disk controllers were in the early days of the PC. Most hardware manufacturers make sure that their hardware and software works with Adaptec products, so choosing an Adaptec is a good first step in trying to make life easier for yourself. Figure 4 lists the model numbers of a range of Adaptec SCSI controllers, showing the functionality provided and the bus architectures supported.

Host adapters offer a range of capabilities. The basic models have no built-in ROM and are suitable only for non-DOS devices like CD-ROM or tape drives which are accessible only through device drivers. Others have a boot ROM which implements the BIOS hard disk functions for SCSI drives. This means that you can boot the PC from a SCSI drive. Without a ROM you must either boot from floppy or from a standard (eg, IDE) hard disk. The most recent host adapters support bus mastering. This gives faster throughput particularly if an ISA bus is used. Hard disk data transfer rates in the region of 2.5 MB/sec are possible.

Hardware Setup

The same considerations apply when installing a host adapter in a PC as any other type of card. First, you should check that there are no conflicts between it and anything else in the machine in terms of the IRQ line, I/O port address or DMA channel used. Change settings if necessary using the jumpers or software setup utility provided, and make a note of your changes ready for when you have to configure your driver software.

If the adapter has its own BIOS ROM and you want to be able to boot from a SCSI hard disk, check that the ROM is enabled. Disable the ROM if you don't want PC BIOS support for hard disks.

Make sure that the memory region used by the ROM does not overlap any other ROM in the system. It is a good idea, if necessary, to change the region used so that it directly adjoins other ROMs in the system. This will help you achieve the most efficient memory management by avoiding fragmentation of upper memory.

You can use the Microsoft Diagnostics (MSD.EXE, as supplied with Windows though it's best to run it outside Windows) utility to check the regions used by other ROMs.

Make a note of the memory region used: you may need to amend the settings used by the memory manager to exclude the ROM region from use.

Check the host adapter's SCSI ID. This should be 7. If the host adapter will not be at one end of the SCSI bus, remove or disable the terminators. And unless you have a good reason for it not to be, make sure terminator power is enabled. The parity checking setting, if present, should only be en-

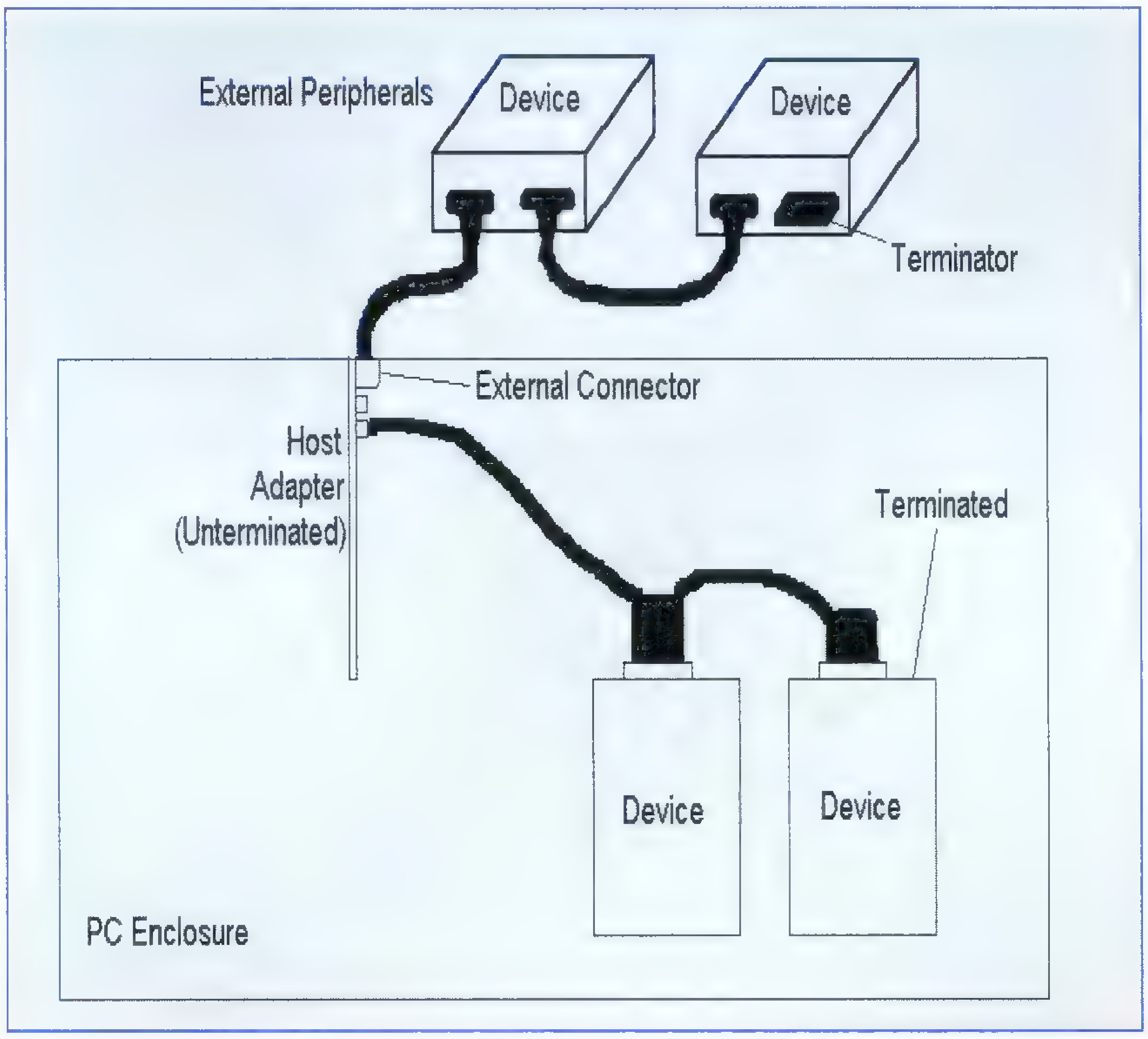

Figure 3 - A mixture of internal and external SCSI devices.

abled if all the devices on the chain support it.

Attach the cables to the host adapter and the SCSI devices. Ribbon cable should be routed as far away from other active components as possible, to prevent noise induction.

You wouldn't intentionally make this mistake, of course, but don't forget to attach power leads to any internal devices. The power to turn the motors and run the electronics is not supplied by the SCSI cable.

For each device, check its SCSI ID, its terminator and its terminator power setting. Terminators must be removed or disabled on each device except those at the end of the cable. Terminator power should also normally be disabled, though you might enable it on one external device if you have several devices in your SCSI chain.

Check the documentation to determine which device IDs should be used. Some device driver software may only work correctly if the device has a specific ID. Hard disks are usually given low numbers: if a hard drive has ID 0 and the host adapter has a ROM which provides BIOS disk support, the system will try to boot from this drive.

Device IDs are often set using DIP switches. Double check which side of the switch is ON and which is OFF: it is all too easy to set the binary inverse.

"Operating systems other than DOS which have built-in SCSI support are the easiest to deal with. As long as your host adapter is supported by the operating system, installing the drivers for the SCSI devices is simply a matter of following the documented procedure for doing so."

Hard Disk Setup

SCSI hard disks can be accessed by a PC in two ways: through the BIOS, or through a device driver. If a device driver is used, the PC must be booted from another drive - either a standard IDE hard disk or a floppy - from which the device driver can be loaded. BIOS services are needed in order to access a SCSI hard disk at boot time.

In general, PCs do not have built-in BIOS support for SCSI - the exceptions are those which have host adapters built in to the motherboard - and so to boot from a SCSI hard disk you must have a host adapter with an on-board BIOS. If there are IDE hard disks in the system, these will take precedence over the SCSI drives. If you want to boot from a SCSI disk then the drive types for the IDE drives, set from the system's Setup utility, must be set to 0 or "Not Installed". The SCSI disk should have both its ID and its LUN set to 0.

Low-level Formatting

Most SCSI drives are shipped already low-level formatted. A few are not. If your hard disk is not low-level formatted then you will need to perform this operation using utilities supplied with the host adapter. On some host adapters these utilities are stored in ROM and accessed by executing a program address using DEBUG. With others, a software utility may be used. Consult the host adapter manual for more specific details.

If the drive is to be used under DOS the next step is to partition it using the DOS FDISK utility. The first partition should be made an active partition if you intend to be able to boot from it. After this, you can format each partition using the DOS FORMAT utility, making sure to specify the /S switch for the bootable partition.

Driver Software

Device drivers will be needed for each type of peripheral attached to the host adapter, apart from hard disks that will be accessed through an on-board BIOS. The precise configuration details will depend on the host adapter and peripherals used, as well as the operating system used, so it is not possible to give specific guidance.

Operating systems other than DOS which have built-in SCSI support are the easiest to deal with. As long as your host adapter is supported by the operating system, installing the drivers for the SCSI devices is simply a matter of following the documented procedure for doing so. As long as the hardware is set up correctly, the device should work.

DOS is more difficult. Because neither DOS nor the PC system architecture includes support for SCSI, setup procedures vary depending on both the hardware and software being used.

	ISA	MCA	EISA
No BIOS, No FDD support	1510		
No BIOS, FDD support	1512		
BIOS, No FDD support	1520		
BIOS, FDD support	1522		
BIOS, No FDD, Bus mastering	1540	1640	1740
BIOS, FDD, Bus mastering	1542		1742

Figure 4 - Key to Adaptec Host Adapter model numbers.

SCSI Devices

"The complexity of many SCSI configurations can make troubleshooting a difficult task. The best approach is to begin by checking the basics. If the PC hangs at boot-up, then there may be either a software or a hardware conflict. Check the I/O ports, IRQ line and DMA channel."

Until recently, SCSI devices were fairly uncommon on PCs under DOS. In many cases, a PC had just a single SCSI device. Typically, SCSI devices were shipped with device drivers for specific host adapters. You simply followed the setup instructions, installed the device driver using a DEVICE= line in CONFIG.SYS, rebooted the system and with luck the new device would then be accessible.

However, these older device drivers don't usually allow you to have different types of device on the same SCSI bus. They are written in such a way that they treat the host adapter as a dedicated controller card. You cannot simply install a second device driver for the other device. The two device drivers will conflict in exactly the same way as, say, a DOS and a Windows comms program would conflict if they were both trying to access the same serial port. This is another reason why some PCs have two or more host adapters installed in them - one for each device.

A better solution is to use device drivers that use a standard interface such as the Advanced SCSI Programming Interface (ASPI) or the Common Access Method (CAM). These two standards are functionally similar. What they do is put a layer of software between the host adapter and the device drivers. Device drivers talk to this software layer instead of directly to the host adapter, and the software arbitrates between the requirements of different devices. This means that, as long as you have the necessary ASPI or CAM drivers, you can take advantage of SCSI's ability to support up to seven devices.

ASPI was originally developed by Adaptec, and indeed the "A" used to stand for Adaptec until the company realised that this was preventing its competitors from adopting it as a standard.

Adaptec's ASPI software is called EZ-SCSI. It can be purchased on its own, but is shipped with most new Adaptec host adapters. There is also wide third-party support for the ASPI standard.

Future Domain has adopted the CAM standard. It produces a similar software kit called PowerSCSI! which supports Future Domain's host adapters. Because of the wider availability of ASPI device drivers, PowerSCSI! includes an ASPI to CAM conversion layer, which in theory allows them to be used where a CAM driver is not available.

To make the most of your SCSI hardware, therefore, you should use software that adheres to one of these two standards. Given the market dominance of ASPI, there are clear benefits for choosing it rather than its rival, CAM.

If you cannot find ASPI or CAM drivers for all your hardware then it is worth investigating CorelSCSI from Corel Corporation, the people behind CorelDraw. This contains ASPI drivers for a wide range of host adapters and devices, as well as diagnostic and other utilities. A more expensive version, CorelSCSI Pro, includes drivers for scanners, as well as ASPI support software for NetWare.

Troubleshooting

The complexity of many SCSI configurations can make troubleshooting a difficult task. The best approach is to begin by checking the basics. If the PC hangs at boot-up, then there may be either a software or a hardware conflict. Check that the I/O ports, IRQ line and DMA channel used by the host adapter don't conflict with anything else installed in the machine. If possible, pull out other expansion cards so that only the bare minimum plus the host adapter remain.

If the card has a ROM, make sure this doesn't overlap any other ROM in the system. Turn off ROM caching, which may be a Setup option. This can interfere with the operation of some cards.

"You cannot mix 8- and 16-bit cards in the same 128 KB region of memory. This means that if you have a 16-bit VGA card with its ROM at the standard address of C0000h, you cannot have an 8-bit card containing ROM or RAM anywhere in the region C0000h to DFFFFh."

"With modern hardware, most SCSI problems are soluble. The SCSI standard is now more tightly specified and this, together with the use of standard ASPI or CAM drivers, has removed many of the incompatibility problems that plagued SCSI users in the past."

Make sure that the driver software is configured to use the I/O, DMA and IRQ settings used by the card. If you changed any of the defaults, you'll probably need to reconfigure the software.

You cannot mix 8- and 16-bit cards in the same 128 KB region of memory. This means that if you have a 16-bit VGA card with its ROM at the standard address of C0000h, you cannot have an 8-bit card containing ROM or RAM anywhere in the region C0000h to DFFFFh.

Make sure that your memory manager isn't trying to use the region used by the host adapter's ROM for upper memory. It must be excluded from both the DOS memory manager (for example, EMM386.EXE X=) and from Windows (EmmExclude=). If you use MS-DOS 6, single-step through CONFIG.SYS to find the point at which the system crashes.

If the system will boot up, look for any messages displayed by the host adapter ROM or its driver software. Many products display the identity of each device it finds on the SCSI chain. If one or more devices are not listed, investigate why. Check that the IDs of each peripheral are different. Make sure at least one device (normally the host adapter) is supplying terminator power.

If the missing device is an external one, is it switched on? A device must be receiving power in order to identify itself to the host adapter. If it is not found when the adapter performs its initialisation processes, the device will be ignored even if it is switched on later.

Some internal devices don't respond to SCSI commands until they have finished their internal diagnostics. The speed with which some fast PCs boot up means that these devices may still be busy when the host adapter looks for them. A sure sign of this is if the device is always missing when you switch on, but it's there if you immediately reboot.

The solution is to try to delay the boot-up process. For example, by enabling a Setup option to perform a full memory check. Or install a device driver for a piece of non-existent hardware, so that the PC spends a few seconds looking for the hardware before timing out and refusing to load the driver.

Check that the host adapter, device drivers and device are compatible. Even the revision number may be significant. For example, older Adaptec CD drivers don't support multi-session drives; older host adapters don't support disks of more than 1 GB capacity under DOS.

The order in which device drivers are loaded can be important: don't tinker with the order in which drivers were installed by an automatic setup routine. Reinstall them if you aren't sure if they have been tampered with or not.

Loading device drivers into high memory can sometimes affect operation. Try loading them into low memory temporarily and see if this cures the problem.

If some SCSI devices are found, but not others, disconnect those that work (and de-install their device drivers) so that you can concentrate on the ones that don't. Make sure that the bus is still correctly terminated. If this makes a missing device appear, put back the others one at a time until you find the conflict.

If there is no conflict of SCSI IDs then it may be that simply altering the position of the device on the bus has been enough to make it appear. This is a sign of a termination problem. Try using different cables, or putting a different device at the end of the chain. Use active termination if possible.

Conclusion

With modern hardware, most SCSI problems are soluble. The SCSI standard is now more tightly specified and this, together with the use of standard ASPI or CAM drivers, has removed many of the incompatibility problems that plagued SCSI users in the past.

As PC users demand access to an increasing variety of storage devices, so the need for SCSI will increase. Hopefully, by following the principles outlined in this article, installing SCSI devices will not be the nightmare that it once was.

PCSA

The Author

Julian Moss is a PC specialist and IT consultant. He is currently preparing a more general article on the basics of SCSI, for publication in a future issue of PCSA.

Understanding VMS

A VAX running VMS is very easy to use. It offers similar facilities to a PC running DOS. While the commands may be different, the underlying structure is similar. If you need to know your way around VMS, or you need to convince someone that you do, let Steve Greenham be your guide.

In 1978 the Digital Equipment Corporation shocked the computing world by announcing a 32-bit operating system. Cynics scoffed. At a time when mainframe RAM was more usually expressed in KB, who would need a system capable of addressing 4.3 GB?

The key to the new OS was its ability to use "virtual" memory, switching data between RAM and disk as required and thus allowing far greater capacity than could be economically achieved using silicon.

A couple of years later Digital delivered and the rest is history. Today VMS is one of the world's most widely used operating systems, running on $5000 workstations up to $5,000,000 mainframes. It has also been ported from the proprietary CISC architecture of the VAX range of minicomputers to RISC and Alpha AXP systems, gaining an Open prefix along the way. Meanwhile VMS's Chief Architect, Dave Cutler, has moved to Microsoft and developed Windows NT (is it mere coincidence that the letters in WNT are each just one away from VMS?). Will NT replace VMS? Time will tell, and Digital is backing both horses. It has committed to support VMS for the next 10 years, so there is plenty of life left in those legacy systems. VMS skills are still much in demand.

Finding Your Feet

On a single-user operating system like DOS you are omnipotent. If you choose to reformat the hard disk no-one will stop you. In VMS your powers will be curtailed and unless you are a system administrator you will only have limited rights to a part of the system.

If you are an administrator, or you're hoping to learn how to become one, then stop reading now - you will need far more training than this introduction can give you. This article is aimed squarely at people who are familiar with DOS and need an introduction to using VMS; copying files, running batch jobs and printing - certainly not reformatting system disks.

Network Topology

VAXes are grouped together in clusters. Each VAX system is called a node and the nodes are linked together with high speed buses. Clusters share resources, such as user disks, printers, tape drives etc. Computing power is also combined because each process can see any CPU in the cluster, thus balancing the workload between nodes.

The node at which you are logged in is called your local node, but from there you can log into other remote notes without having to log out of your current session. You use the command **SET HOST** to specify the remote node to which you wish to log in.

For example, typing

```
$ SET HOST JUPITER
```

will start the login procedure for the node called "Jupiter".

Devices attached to VAX computers fall into two broad categories, namely mass storage devices like disk drives and magnetic tape drives, and record-oriented devices like terminals, printers and card readers (yes, card readers. VMS has been around that long!).

These devices are identified by cryptic physical device names, but that's the system administrator's problem. Logical device names allow users to refer to devices by names that are more memorable, such as "FREDS PRINTER".

Getting Started

You may be accessing a VAX from a dumb terminal or using a terminal emulator. In either case it is important that it has been set with the correct connection parameters. If in doubt try 9600 bps at 8N1. A serial connection is usually routed via a "DECserver" which has several serial inputs and an Ethernet connection to the Local Area Network (LAN). In this case, pressing Return elicits a **Local>** prompt. The command to connect to a given VAX is **CONNECT**, which can be abbreviated to C. For example to connect to the VAX2 cluster you would type:

Local> C VAX2

You will then be prompted for your user identity and password. The password is not displayed on the screen. A PC with a terminal emulator, using the serial port, will connect in exactly the same way as a dumb terminal. The PC or terminal may also have its own Ethernet card and connect using LAT (Local Area Transport) or TCP/IP.

One other thing to be aware of is that the VT family of terminals have different key mappings to the standard 102 key PC keyboard. If you are using a PC you may find there is a keyboard overlay to identify esoteric keys, such as the "Gold" key used in the VMS editor.

Accounts And Passwords

When you are given access to a VAX system you are issued with a user name. The naming convention is likely

to depend on the number of potential users of a system because each name must be unique. It may be based on your name or initials and possibly combined with a number, eg, FRED, FASMITH or FAS27. This name is called your user account and is used by the system to track your use of devices, ownership of files etc. Your user name is not secret - in fact it will probably be the name under which other people email you.

When you log in you are asked for your user name and your password. The first time you log in you should change the password provided by your system manager to one of your own choosing. A wise system manager will have pre-expired the password to ensure that you do. By default, passwords must be changed every 180 days, although this can be set to a shorter time. Passwords have a minimum length specified by the system manager and a maximum length of 31 characters.

Change your password with the **SET PASSWORD** command. The **/GENERATE** qualifier produces a list of five random passwords, which are claimed to be pronounceable. Use of the password generator can be made compulsory by the system manager. Passwords can use letters, numbers, dollar signs and underscores, and are not case-sensitive. When you set a password on one node, it is replicated on all other nodes of the same cluster. For example:

```
$ SET PASSWORD/GENERATE
Old password:

dilebco     di-leb-co
ukwauxmat   uk-waux-mat
eoggafroa   e-og-ga-froa
gohaust     go-haust
revzowe     rev-zowe
```

Choose a password from this list, or press RETURN to get a new list.

Accounts can be Captive or Interactive. A captive account only allows the user limited access to the system, typically through menus, and no direct access to VMS. Interactive accounts give you access to the system software.

Digital Command Language

You interact with VMS using Digital Control Language, DCL. The default DCL prompt is a $ but can be changed with the **SET PROMPT** command. The system displays a **$** when it is awaiting a command.

The syntax of a **DCL** command line is:

```
Command Parameter, Parameter
/Qualifier=Value or Keyword
```

Every command line must have at least one Command and one Parameter (eg, **SHOW TIME**). Additional parameters may be required for some commands. Qualifiers are optional and act to modify the command. Some qualifiers take a value such as a number, string or DCL keyword. The command line:

```
$RENAME *.TXT;* *.BAK;* -
_$/MODIFIED /SINCE=-
_$YESTERDAY
```

has a prompt ($), a command **(RENAME)**, two parameters (the source and destination file specifications) and two qualifiers (/MODIFIED and /SINCE). The second qualifier takes a value, in this case a DCL keyword (YESTERDAY). If you omit any required parameters the system will prompt you for them. Long command lines can be split by ending the first part with a hyphen; you are then prompted for the rest with the DCL continuation prompt (_$).

In this introduction to VMS it is impossible to cover every **DCL** command and all their qualifiers. Instead I have tried to cover the most frequently used commands in sufficient depth that you can apply the information to other commands as you discover them.

Many qualifiers, such as those for file dates and protection, are the same for a number of commands. If you understand the syntax for COPY you will be able to make a pretty good guess at RENAME or DELETE.

Like MS-DOS, **DCL** commands can be grouped together into batch files, which are called Command Procedures and given a .COM extension. Command procedures can be executed interactively by typing the filename preceded by an "@" sign:

```
$ @CLEANUP.COM
```

When you write a command procedure you must you must include a **$** at the start of every command line - a point often forgotten by DOS programmers. Comments have an exclamation mark and the file should be terminated with an **EXIT** command. For example:

```
$ ! CLEANUP.COM - Fred's file
$ PURGE *.TXT
$ EXIT
```

LOGIN.COM runs each time you log in so you can use it to DEFINE logical names as a personal shorthand to refer to disks, directories or even files. For example, if Fred stores his

"Use of the password generator can be made compulsory by the system manager. Passwords can use letters, numbers, dollar signs and underscores, and are not case-sensitive. When you set a password on one node, it is replicated on all other nodes of the same cluster."

Understanding VMS

"A major difference between DOS and VMS is that VMS directories are themselves files, which have a .DIR extension. These directory files contain references to the physical location of the other files on the disk."

sales figures in a new subdirectory each month then you might use

$ DEFINE FRED_SALES [FASMITH.-SALES.1994.JULY]SUMMARY.DAT

to refer to his current figures.

You can also define symbols to represent **DCL** commands; for example, a DOS change directory.

$ CD :== SET DEFAULT

Control Keys

You can interrupt commands by using control key combinations, as follows:

CTRL-S Suspends screen output.
CTRL-Q Resumes screen output.
CTRL-C Cancels command processing during entry.
CTRL-Y Interrupts current command once it is running.
CTRL-Z Exit current command, indicates end of file.
CTRL-T Interrupts command to display process statistics.

You can also recall previously-typed commands from a buffer of 20 command lines by using the Up and Down keys or CTRL-B.

Disk Management

Each user is allocated a certain amount of disk space, called their quota. As files and directories are created this quota is used up and eventually you run out of space. At this point you can either delete some files or plead with your system manager for more quota. To see how much disk space you have used, and how much remains use the **SHOW QUOTA** command.

Directories

Each physical disk has a main directory called the Master File Directory (MFD). It is named [000000]. The MFD contains a list of User File Directories (UFDs) - essentially a list of the registered users of the system. Users normally log in to their UFD.

A major difference between DOS and VMS is that VMS directories are themselves files, which have a .DIR extension. These directory files contain references to the physical location of the other files on the disk. Subdirectories are directory files which are neither MFDs or UFDs and can contain references to files or to other subdirectories, up to seven layers deep. An example of a directory tree is shown in Figure 1.

User directories are created by the system administrator and, for a given user, only the system administrator and that user should have WRITE access rights. If anyone else has WRITE access to your UFD, they will be able to work their way down to any subdirectory, removing any protection you have set.

Creating Directories

You can create subdirectories in any directory to which you have WRITE access using the **CREATE/DIRECTORY** command. Specify the name of the subdirectory in square brackets preceded by a period.

$ CREATE /DIRECTORY [.1994]

This will create a subdirectory called 1994 below the current directory (the DEFAULT directory). You must specify the full path if you wish to create a subdirectory in a directory to which you are not currently attached. The /LOG qualifier asks the system to confirm that the directory was successfully created. You can use the /VERSION_LIMIT=n qualifier to

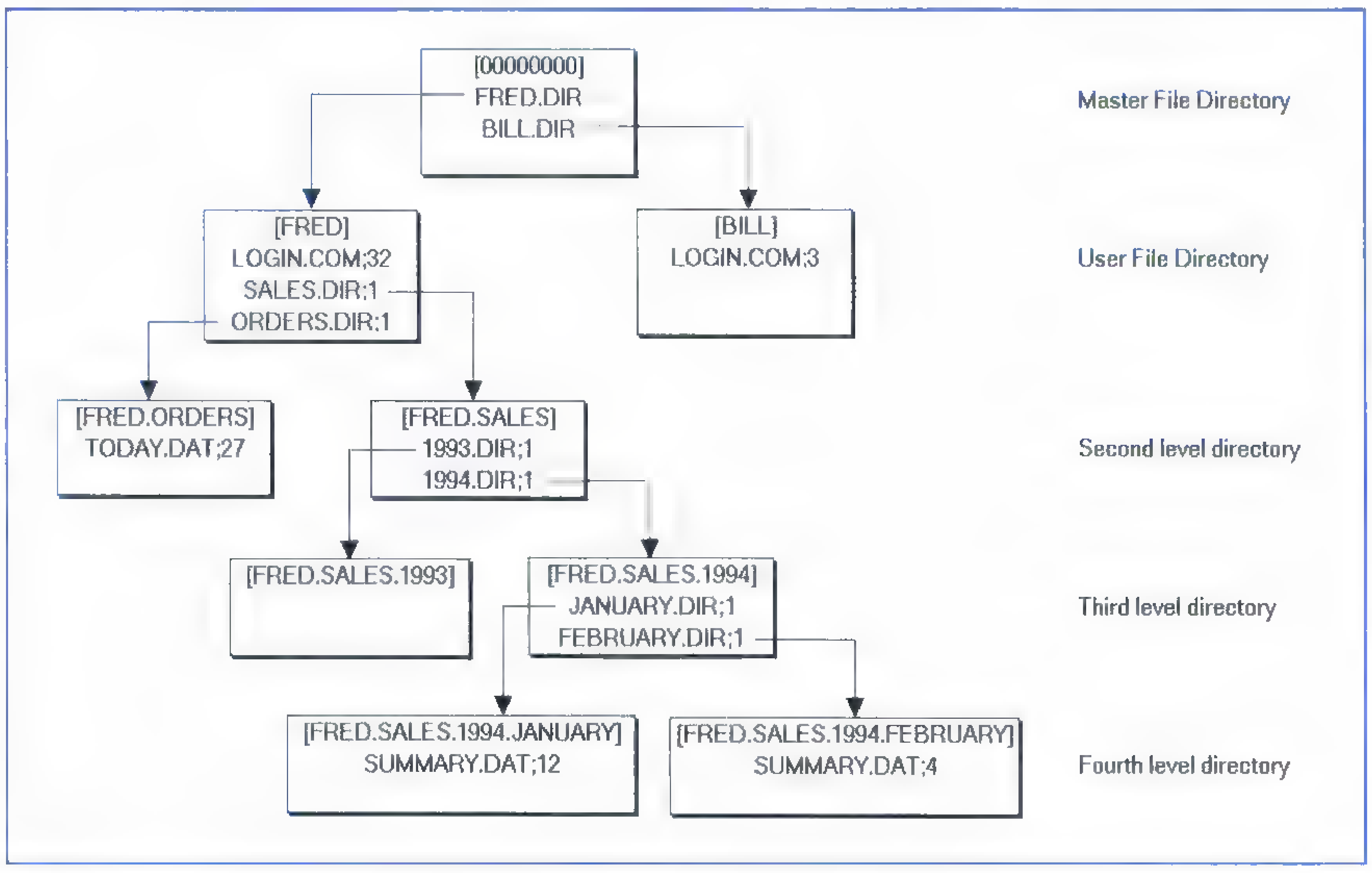

Figure 1 - Example directory tree.

set a maximum number of versions of a file that can exist in the directory (between 1 and 32,767) and the /PROTECTION qualifier to set access rights. In both cases, the default is that of the parent directory.

```
$ CREATE/DIRECTORY/LOG /VERSION_LIMIT=5[FRED.SALES.1994.JULY]

%CREATE-I-CREATED, VAX2:-[FRED.SALES.1994.JULY] CREATED
```

Directory name conventions are the same as those of filenames. Remember you can only have seven levels of subdirectory below a top-level directory.

Moving Between Directories

At any given time, VMS is maintaining a pointer to your default directory and any commands that do not explicitly refer to a directory are assumed to refer to that. To change the current directory you set a new default:

```
$ SET DEFAULT [FRED.SALES.1994.-MAY]
```

You can check where you are with the **SHOW DEFAULT** command:

```
$ SHOW DEFAULT
VAX2:[FRED.SALES.1994.MAY]
```

You can also move up the directory tree using:

```
$ SET DEFAULT [-]
```

and down the tree by specifying just the subdirectory, preceded by a period:

```
$ SET DEFAULT [.MAY]
```

Displaying Directories

Use the **DIRECTORY** command to display the contents of any directory to which you have READ access. You can use wildcards to display files matching a particular pattern.

VMS stores far more information about files than does DOS. As well as the file size and creation date VMS

Security And Ownership

/ACL	Display the access control list for each file.
/BY_OWNER[=uic]	Display files owned by the specified user identification code. If no UIC is specified then displays files owned by the current process. You could use this in a shared directory to list files you had created.
/OWNER	Display the user identification code of the file owner.
/PROTECTION	Display file protection information.
/SECURITY	Combination of /ACL, /OWNER and /PROTECTION.

Dates And Times

/BEFORE[=time]	Display files created before the specified time or date. Dates and times are in the format 1994-JUL-23:15:32:27. You can use the keywords TODAY, TOMORROW and YESTERDAY in place of a specific date. The default is TODAY.
/SINCE[=time]	Display files created after the specified time or date.
/BACKUP	Modifies /BEFORE and /SINCE to specify date backed up rather than date created.
/EXPIRE	Modifies /BEFORE and /SINCE to specify date file is set to expire.
/MODIFIED	Modifies /BEFORE and /SINCE to specify date the file was last modified.
/DATE[=options]	By default display the creation date of each file. Use the BACKUP, EXPIRE and MODIFIED options to show the other dates, or ALL to show all four dates.

Display Format And Output Device

/BRIEF	The default. Display the filename, type and version number.
/FULL	Display everything the system knows about the file - and it's a lot!
/COLUMNS=n	Number of columns of filenames in a brief display, between 1 and the number that will fit on the screen.
/VERSIONS=n	Defines the number of versions shown of each file. The default is all versions, 1 displays only the most recent version.
/OUTPUT[=filespec]	Redirects output to a file.
/PRINTER	Redirects output to the system printer.

Figure 2 - Useful directory qualifiers.

Understanding VMS

stores the file owner, access control list, backup, creation, expiration and modification dates, and version information. There are qualifiers to show or hide all of this information, as well as to format the display and redirect it to a printer. Some of the more useful qualifiers are shown in Figure 2.

File Specifications

VMS file names have three components: a name, type and version. Together these form the File Specification. The format is:

filename.- type;version

for example:

LOGIN.COM;32

Each time the file is edited VMS saves the original file and creates a new file with an incremented version number. File names can be up to 39 characters long and can contain letters A-Z, numbers 0-9 and underscore, hyphen and dollar symbols. Names must not start or finish with a hyphen, or start with a dollar. VMS file names are not case-sensitive.

File types are, like DOS, optional and can be from 0 to 39 characters long, with the same restrictions as for file names. If you don't specify the file type when you refer to a file then DCL adopts the following defaults:

CLD	Command description file.
COM	Command procedure file.
DAT	Data file.
DIS	Distribution list file for the **MAIL** command.
DIR	Directory file.
EDT	Startup command file for the EDT editor.
EXE	Executable program file.
HLP	Input source file for HELP libraries.
JOU	Journal file created by the EDT editor.
LIS	Listing file (created by a language compiler or assembler).
LOG	Batch job output file.
MAI	Mail message file.
MEM	Output file created by Digital Standard Runoff.
OBJ	Object file created by a language compiler or assembler.
RNO	Input file for Digital Standard Runoff.
SIX	Sixel graphic file.
TJL	Journal file created by the VAXTPU and ACL editors.
TMP	Temporary file.
TPU	Command file for the VAXTPU editor.
TXT	Input file for text libraries or **MAIL** command output.

File versions can be preceded by a period or semicolon when you type them, but are displayed with a semicolon. If you refer to a file and don't specify a version number then VMS will use the file with the highest version number.

The flexibility allowed by the file name, type and version can result in very long file specifications. For example:

MY_SECOND_FINANCE_REPORT .STOCKS_AND_SHARES_IN_$_UN ITS;21365

is perfectly legal - but a real pain to enter. Make file specifications long enough to be meaningful and short enough to be usable.

Wildcards

VMS has four wildcard characters used to create patterns for filenames and directories. The * and % wildcards are similar to * and ? in DOS, while the ellipsis (...) and hyphen (-) wildcards are used to refer to the directory tree only and do not have a DOS equivalent.

The * wildcard replaces either part or all of a field, in the directory, name and type parts of the file specification. It can also be used to match the entire version field - but not part of it. If the semicolon and version fields are omitted, then all versions are shown. Here are some examples:

$ DIRECTORY *.TXT
Lists all .TXT files in the directory.

$ DIRECTORY *.TXT;*
Same as above.

$ DIRECTORY BOOK*.TXT;*
Lists all .TXT files whose name starts with BOOK, including BOOK.TXT itself.

$ DIRECTORY BOOK*.*;3
Lists all files of all types whose name starts with BOOK and are at version 3.

$ DIRECTORY BOOK.TXT;*
Lists all versions of the file BOOK.TXT.

$ DIRECTORY BOOK.TXT;3*
This is not allowed.

If the semicolon is included, but the version field is not specified, then VMS defaults to the latest version. The same happens if it is specified as 0.

For example, DIRECTORY CHAPTER*.TXT; or DIRECTORY CHAPTER*.TXT;0 lists the latest ver-

*"VMS has four wildcard characters used to create patterns for filenames and directories. The * and % wildcards are similar to * and ? in DOS, while the ellipsis (...) and hyphen (-) wildcards are used to refer to the directory tree only and do not have a DOS equivalent."*

DOS Synonyms

The worst thing about moving from one operating system to another is that you probably know what you want to do, but you don't know the syntax to do it in the new environment. This table shows common MS-DOS commands and their VMS equivalents.

MS-DOS Command	VMS Equivalent	Function	Syntax
assign	ASSIGN	**DOS:** Redirect requests for disk operations on one drive to another. **VMS:** More powerful, can be used to substitute a logical name for any string.	This example assigns the logical name FRED to the UID of Fred A. Smith on VAX2. ASSIGN VAX2:[FASMITH] FRED
attrib	SET FILE	Display or change file attributes.	SET FILE name /qualifier=value
backup	BACKUP	Backup files to a SAVE SET.	Handled by the system managers
chdir	SET DEFAULT	Change directory.	SET DEFAULT [FRED.DATA
command	SPAWN	Starts a new subprocess.	SPAWN PAYROLL
copy	COPY	Copies files.	COPY FRED*.TXT *.BAK
date	SHOW TIME	Display or change the date (can't change in VMS).	SHOW TIME
del	DELETE	Delete files.	DELETE *.BAK;*
dir	DIRECTORY	Display list of files in a directory.	DIRECTORY *.TXT;*
doskey	RECALL	Maintain keyboard command stack.	RECALL /ALL
edit	EDIT	Start the full screen editor.	EDIT/EDT or EDIT/TPU
fc	DIFFERENCES	Compares two files.	DIFFERENCES file1 file2
find	SEARCH	Searches for a string in a file.	SEARCH *.DOC conclusions
for	GOTO	Loop in batch files or command line.	VMS only looping construct is IF ... THEN GOTO label:
help	HELP	Online help.	HELP topic
mem	SHOW MEMORY	Display the amount of free memory.	SHOW MEMORY /PHYSICAL_PAGES
mkdir	CREATE/DIR	Create a subdirectory.	CREATE/DIRECTORY newstuff
more	/PAGE	Pages screen listings	TYPE FRED.TXT /PAGE
nlsfunc	NCS	Country specific information for national language support.	NCS file-spec
path	DEFINE	Set search path for executables.	DEFINE myprogs [FRED.EXES], - [FRED.PROGS]
print	PRINT	(Background) prints a file.	PRINT FRED.TXT
prompt	SET PROMPT	Change the command prompt.	SET PROMPT =What now?
rename	RENAME	Rename a file.	RENAME FRED.TXT BERT.TXT
restore	BACKUP	Opposite of backup.	
rmdir	DELETE name.DIR;*	Delete a directory.	DELETE SALES.DIR;* n.b. Directory must be empty
set	SET	Set environment variables.	There are a lot of SET commands
sort	SORT	Sorts an input and pipes it to an output.	SORT CHAOS.DAT ORDER.DAT
subst	DEFINE	Associates a directory path with a drive letter.	No drive letters in VMS but use DEFINE C [FRED.DATA.STUFF] instead
time	SHOW TIME	Displays/sets system time (can't change in VMS).	SHOW TIME
type	TYPE	Display a file.	TYPE LIST.TXT
ver	SHOW SYSTEM	Display version number.	VMS Version number is included at the top of a huge listing!

Figure 3 - Common MS-DOS commands and their VMS equivalents.

Understanding VMS

sions of all .TXT files whose name starts with CHAPTER.

Negative version numbers can be used to specify versions preceding the latest version. For example, DIRECTORY CHAPTER*.TXT;-2 lists the CHAPTER files two versions before the latest version. So if the latest version of CHAPTER3 is version 36 it will list CHAPTER3.TXT;34.

The % wildcard works in the same way as the DOS ? wildcard - it replaces individual characters, so DIRECTORY CHAPTER3%.TXT;0 lists the latest versions of CHAPTER30.TXT to CHAPTER39.TXT, but not CHAPTER3.TXT or CHAPTER300.TXT.

The % wildcard can't be used in the version part of the field.

Wildcards can also be used in the directory field. DIRECTORY [*]LOGIN.COM; lists the latest versions of every user's LOGIN.COM on the current node - assuming you have read access to it.

DIRECTORY [FRED.SALES.-199%.J*]*.DAT; lists the latest versions of all data files in the January, June and July subdirectories of every year between 1990 and 1999.

The ... wildcard recurses down the directory hierarchy, from the current default directory or a specified directory. For example:

DIRECTORY [...]SUMMARY.DAT lists all SUMMARY.DAT files in the current directory and all subdirectories below it.

DIRECTORY [...JUNE]SUMMARY.-DAT searches down the directory tree and lists all SUMMARY.DAT files in directories that end in .JUNE.

DIRECTORY [...SALES...]SUMMARY.-DAT lists all SUMMARY.-DAT files in the SALES subdirectory, and all subdirectories below it.

DIRECTORY [...SALES...JUNE]SUMMARY.DAT lists all copies of SUMMARY.DAT in all subdirectories ending in June, which are below the sales subdirectory.

The "-" wildcard refers to the parent of the current directory so if you are in FRED.SALES then DIRECTORY [-] lists all files in the FRED user directory. And DIRECTORY [-.COSTS] lists all files in the FRED.COSTS subdirectory.

Use multiple hyphens to refer to directories more than one level above you. So DIRECTORY [--] lists all files in the directory two levels above you.

Attempting to point above the Master File Directory raises an error.

Viewing Files

The **TYPE** command displays the contents of the specified file or group of files on the current output device (usually the screen). Output can be paged by giving the /PAGE qualifier (like the | MORE pipe in DOS). You can interrupt the display by hitting CTRL-Y, and resume by typing CONTINUE.

TYPE SUMMARY.DAT /PAGE displays the SUMMARY.DAT file page by page.

Printing Files

The thing to remember when printing from VMS is that there are likely to be a lot of different printers, and it is easy to send your document to the wrong one. There may be a high-speed wide-carriage line printer that is ideal for program listings, a 600 dpi laser printer for documentation and so on.

Each printer has its own print queue to which jobs are submitted, then executed in order of priority and submission. Each job is assigned a job number. One queue is defined as the system default (SYS$PRINT) and will get any print jobs not assigned to a specific queue. For example, PRINT MYSTUFF.TXT prints MYSTUFF.TXT on SYS$PRINT. You'll get a message such as "Job MYSTUFF.TXT (queue SYS$PRINT, entry 197) pending on SYS$PRINT". Use the **SHOW QUEUE** command to list the printer queues available to you. You can then submit jobs to other queues by specifying it in the /QUEUE parameter.

PRINT MANUAL.DOC /QUEUE=linotype prints MANUAL.DOC on the queue with the logical name linotype.

If you have a large print job, and it isn't urgent, then it is good manners to set a low priority on it using the /PRIORITY=n qualifier. The value of n can be between 0 and 255, although a lower limit may have been set by the system administrator. You may also be able to set a higher priority than the default value to jump the queue with a rush job.

If you wish to check the status of your print jobs then use the **SHOW ENTRY** command. For example, SHOW ENTRY 197 shows the status of job 197, ie, whether or not it has been printed yet. If you do not specify a job number then all of your jobs, on all print queues, are listed.

If your job is still pending, and you wish to cancel it, then use the **DELETE/ENTRY** command. For example, DELETE/ENTRY=197 cancels job 197 on SYS$PRINT.

Copy, Move And Rename

COPY OLDSTUFF.TXT NEWSTUFF.DAT copies a file within a subdirectory, giving it a new name. COPY OLDSTUFF.TXT [BILL.DATA] copies the file to a different subdirectory.

RENAME OLDSTUFF.TXT NEWSTUFF renames the file to NEWSTUFF.TXT, while RENAME OLDSTUFF.TXT [BILL.DATA] moves the file to the specified subdirectory.

If the destination file already exists when a file is copied, moved or renamed then the new file will be given a version number one higher than the existing file. Otherwise it will be given the version number of the source file.

Remember that directories are also files? You might expect that copying [FRED]SALES.DIR to [BILL] would produce a copy of Fred's entire sales directory hierarchy under Bill's user file directory. This does not happen, SALES.DIR is copied as an empty directory file. It has the same effect as typing CREATE/DIRECTORY [BILL.SALES].

Deleting Files

Deleting files is very similar to DOS. The only thing to bear in mind is that the wildcard syntax is different, and you must specify version numbers. You can use the date and time

qualifiers discussed previously to delete files between particular dates. For example, if you had a really bad day coding then DELETE *.C;* /SINCE=TODAY will delete all versions of C source files created today.

Purging Files

As you can have up to 32,767 versions of each file in a directory, it can get very overcrowded. The **PURGE** command deletes all but the most recent version of the specified files. Be careful, because the default is to purge all but the most recent versions of all files in the directory.

For example, PURGE *.TXT deletes all old versions of files with the file type TXT.

Security

Security in VMS systems is based on two mechanisms, namely User Identification Codes (UICs) and Access Control Lists (ACLs). UIC security handles broad groups of users, whereas ACLs allow finer control but require more administration. Security is set on all system objects, but we will restrict the discussion to user files.

UIC security is based on four groups of users:

SYSTEM	Users with system management rights.
OWNER	The user who created the file.
GROUP	Members of the same workgroup as the owner.
WORLD	Everybody on the system, including all of the three previous groups.

These are abbreviated to S, O, G and W.

There are four rights within each group, abbreviated to R, W, E and D. These are:

READ	View, print and copy a file.
WRITE	Change a file and save it back.
EXECUTE	Run an executable file.
DELETE	Delete the file.

The default security level, given by the system to all files in your user directory, is (S:RWED, O:RWED, G:RE, W). The point to be aware of is that any other members of your group can list and execute the files in your user directory unless you protect them. You can change the protection of an object either when you create it, by using the /PROTECTION qualifier or after you create it, by using the **SET PROTECTION** command. For example,

CREATE/DIRECTORY/PROTECTION=(S:RWED, O:RWED, G, W) SECRETS

and

SET PROTECTION- =(S:RWED, O:RWED, G, W) SECRETS.DIR.

If you do not specify protection when you create a file, rights are inherited either from the previous version or, for a new file, from the directory in which it is created.

Access control lists provide a much more sophisticated form of protection, which can be based on UIC information including user name, special groups created in a system rights database (eg, for multi-departmental project teams) or by system-defined identifiers such as DIALUP sessions. Thus a named user might be given access to a directory when running a batch job which they couldn't use interactively.

Editors

The standard VMS editor is called EDT, although there may be other Digital and third-party editors present on your system. Invoke EDT by typing **EDIT/EDT filename**. If the filename doesn't already exist then it is created. Digital produce a computer based training package for EDT. Try typing EDTCAI to see if it is present on your system.

EDT has two modes; a command mode, whose prompt is an *, and a full-screen editing mode. Type **C** to change from command mode to editing mode, and Ctrl-Z to interrupt editing mode to enter a command. The **EXIT** command saves the file and exits the editor, while the **QUIT** command exits without saving.

Other editing functions are achieved using the EDT keypad, which usually maps on to the keyboard's numeric keypad. This is fine on a dedicated Digital keyboard, or a VMS compatible PC keyboard, such as the KEA ones, but it is impossible to predict the key assignments of a standard PC keyboard. This is why I prefer to use the public domain editor MicroEMACS. MicroEMACS is available on all major platforms so is recommended if you need to support a number of different types of computer and only want to learn one text editor.

Getting More Help

VMS's help system is very easy to use. At the $ prompt type HELP and, optionally, the keyword you want help on. A list of topics or subtopics is displayed and you enter the name of the one you are interested in. To go back a level press return, keep pressing return to exit HELP. Many systems also have a computer based training module called VMSCAI.

VMS For DOS Users

The worst thing about moving from one operating system to another is that you probably know what you want to do, but you don't know the syntax to do it in the new environment. Figure 3 shows common MS-DOS commands and their VMS equivalents.

PCSA

The Author

Steve Greenham works as an IT analyst at a major international company.